GPU use at the University of Basel

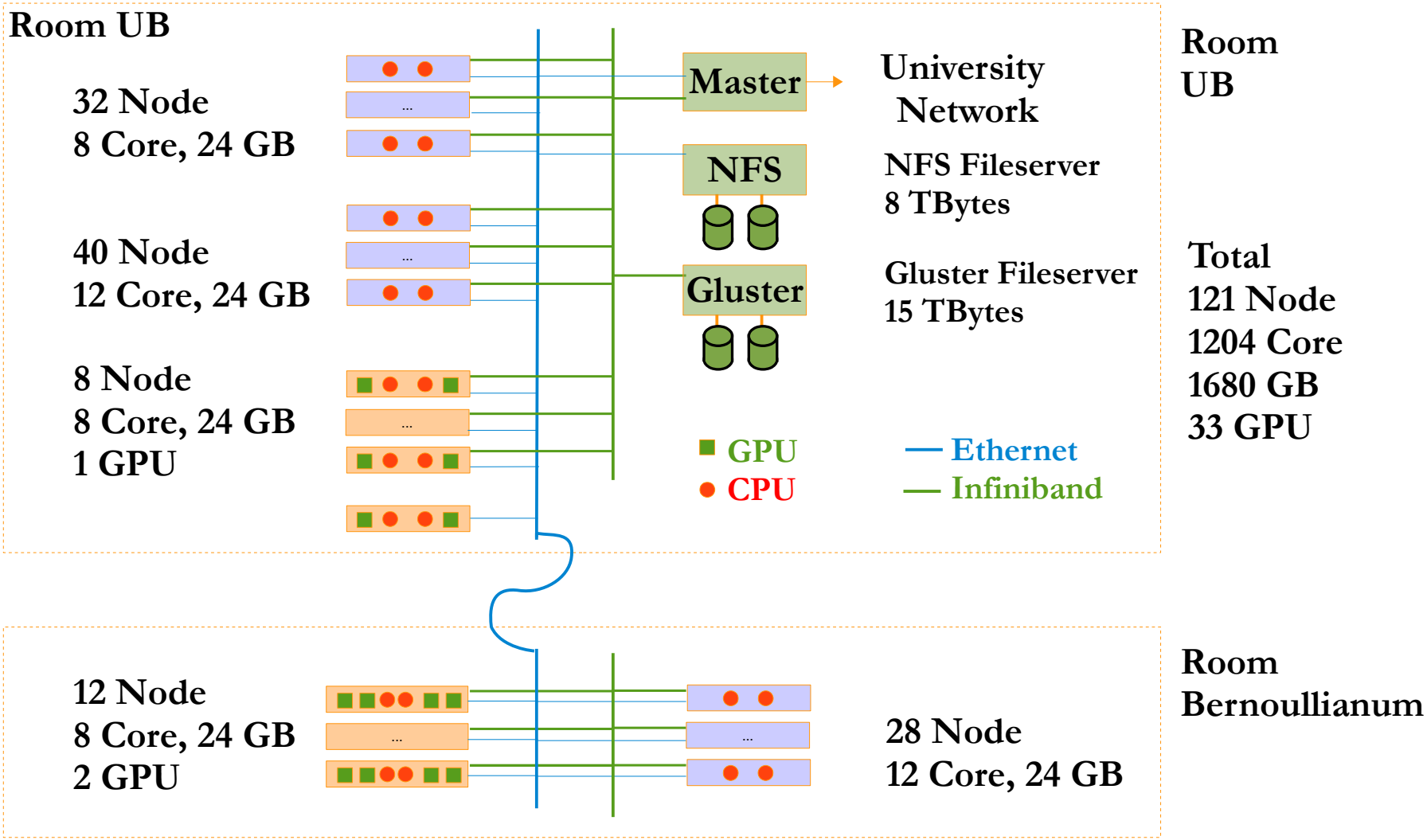
Martin Jacquot
University of Basel (URZ)

HPC-CH Meeting / University of Basel / October, 27th 2011

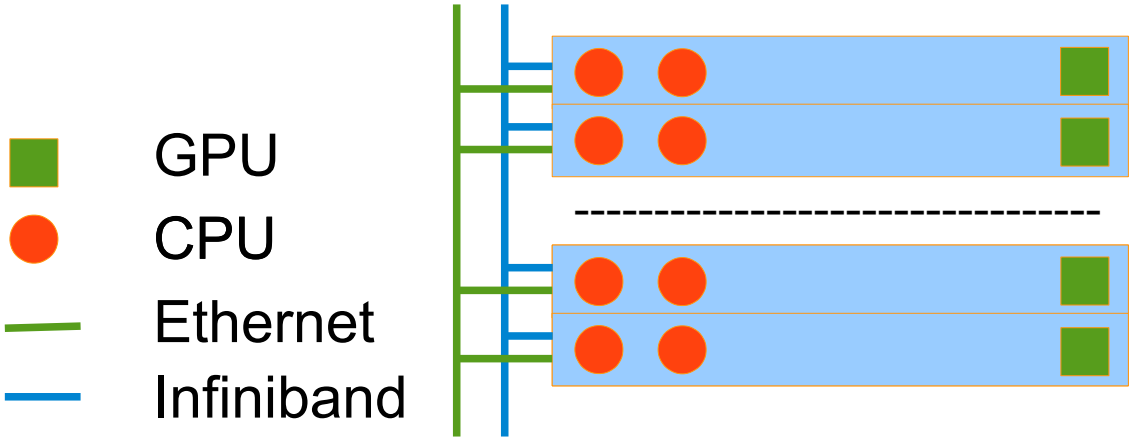
- Motivation
- Cluster infrastructure
  - Cluster architecture
  - GPUs environment, Software environment
- GPU integration
  - Batch-queue System
  - Monitoring, Reporting
- Results
  - Performance, Efficiency
- Conclusion

- Motivation
  - Increase the computing power
  - Meet future needs
  - Contribution to the BigDFT Project
  - Lower acquisition cost
  - Reduce the power and the cooling consumption
  - Try a new technology

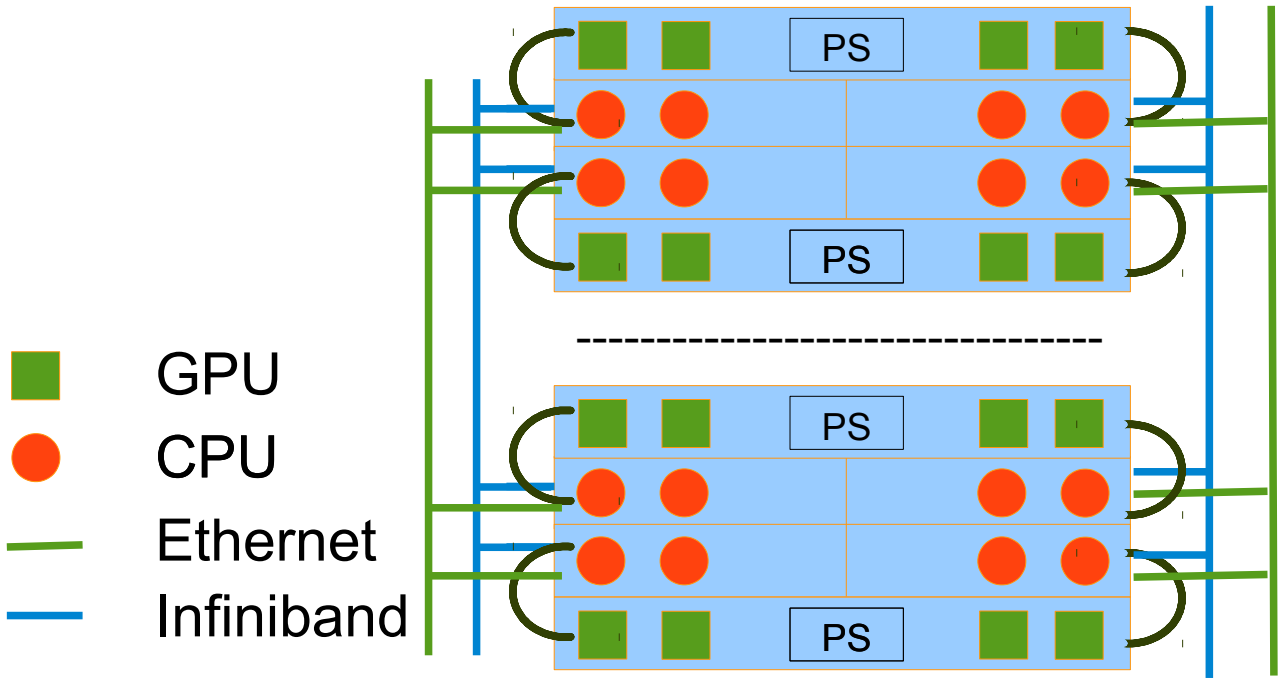
- Cluster Architecture



- Environment URZ with GPUs
  - 8 Nodes (24GB memory)
    - 8 GPUs Tesla C1060
    - 64 Cores

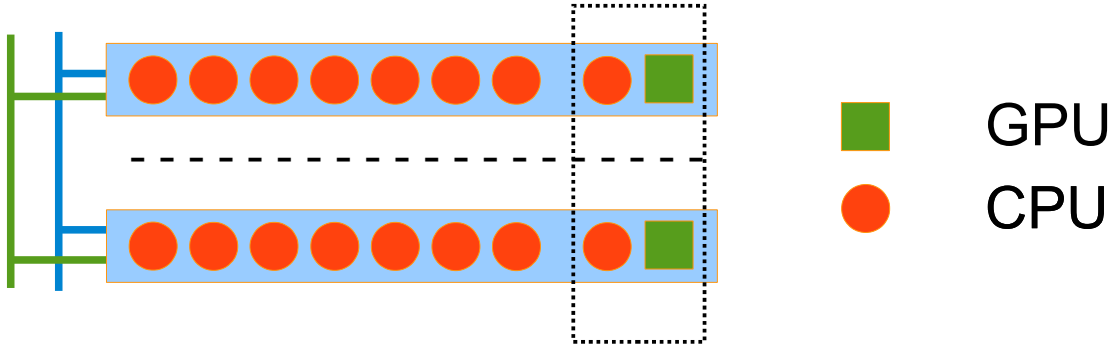


- Environment Physik with GPUs
  - 12 Nodes (24G memory)
    - 24 GPUs Tesla C1060
    - 96 Cores



- Software
  - Operating System : CentOS 64-bit V5.5
  - Queuing system : Open Grid Schedule V6.2u5
  - Compiler, Libraries : Intel, Gnu, PGI
  - Parallel libraries : openmpi, OpenMP
  - Cuda Toolkit V3.2
  - OpenCL V1.0
  - GPU Application : Namd2, BigDFT, Matlab

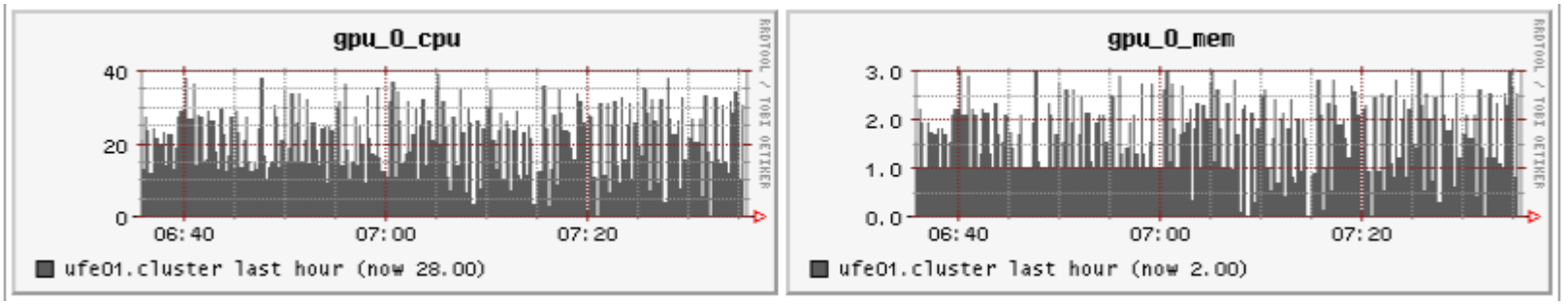
- SGE integration
  - Each GPU is tied to one Core



  - Create a forced complex **urz_gpu=1**
  - Create a queue **gpu.q** with one slot
  - Assign the forced complex with the queue
  - Instruct user to use “qsub -q gpu.q -l urz_gpu=1”
  - Create submit script for the applications

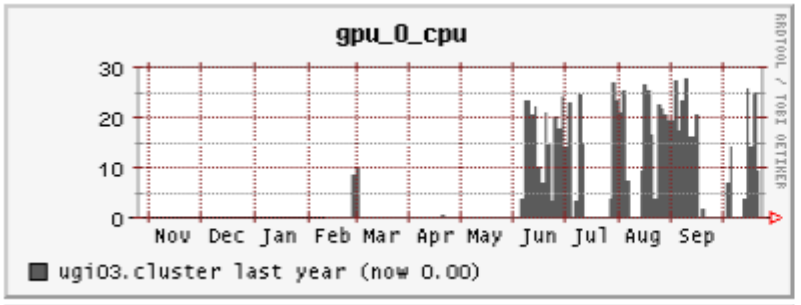
- Monitoring
  - New system management interface with Cuda V3.2
  - x = `nvidia-smi -a -l -i 1 | grep … | awk ...`
  - gmetric --type int32 --name gpu_0_cpu -value $x

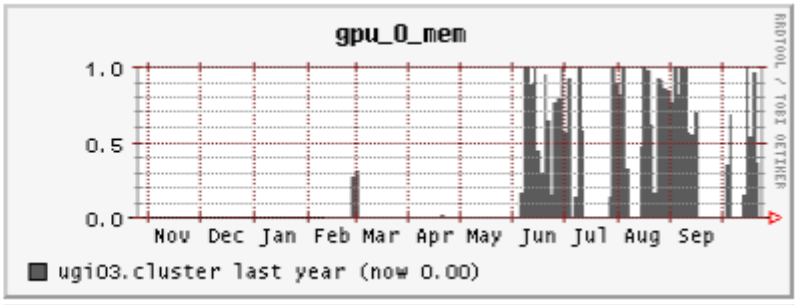
GPU hourly Overview of node ufe01



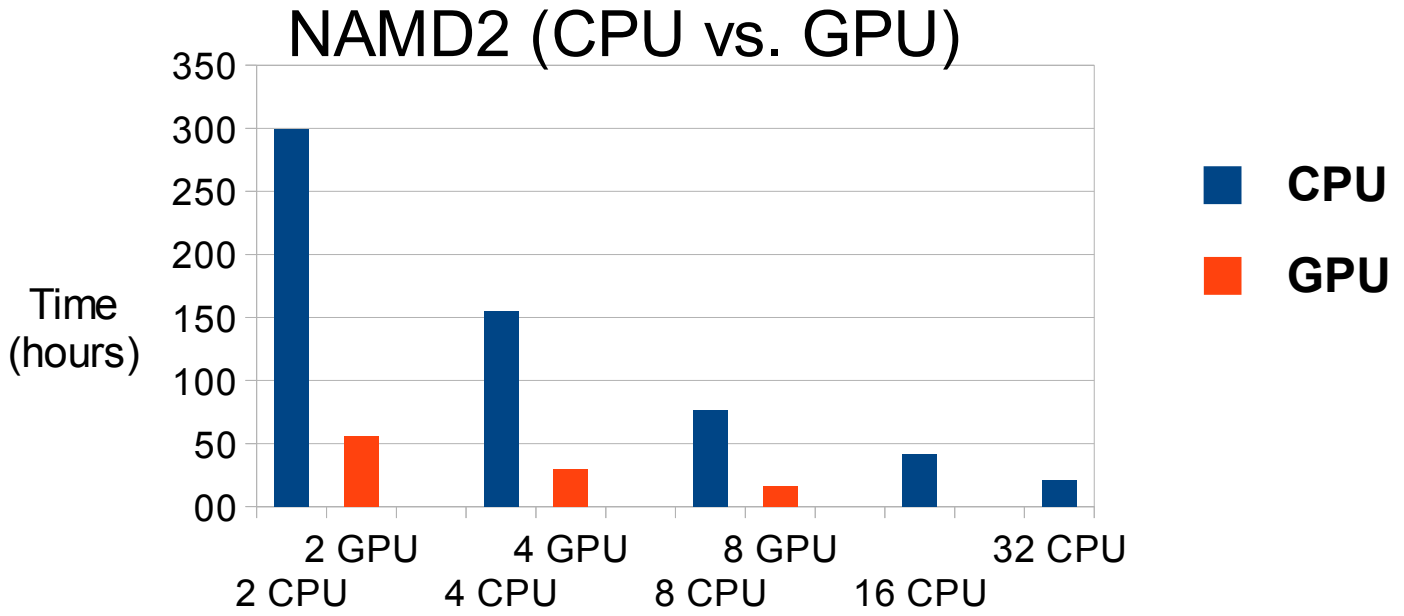
- Reporting
  - No explicit reporting in SGE
  - No GPU, no memory usage accounting
  - Report with qacct -q gpu.q
  - Get GPU usage with ganglia

GPU yearly Overview of node ugi03



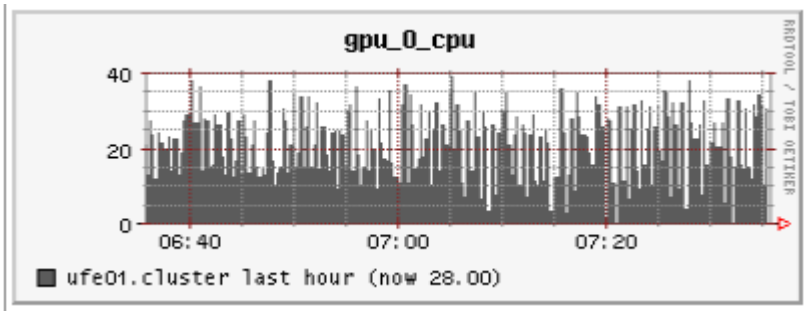


- Runtime

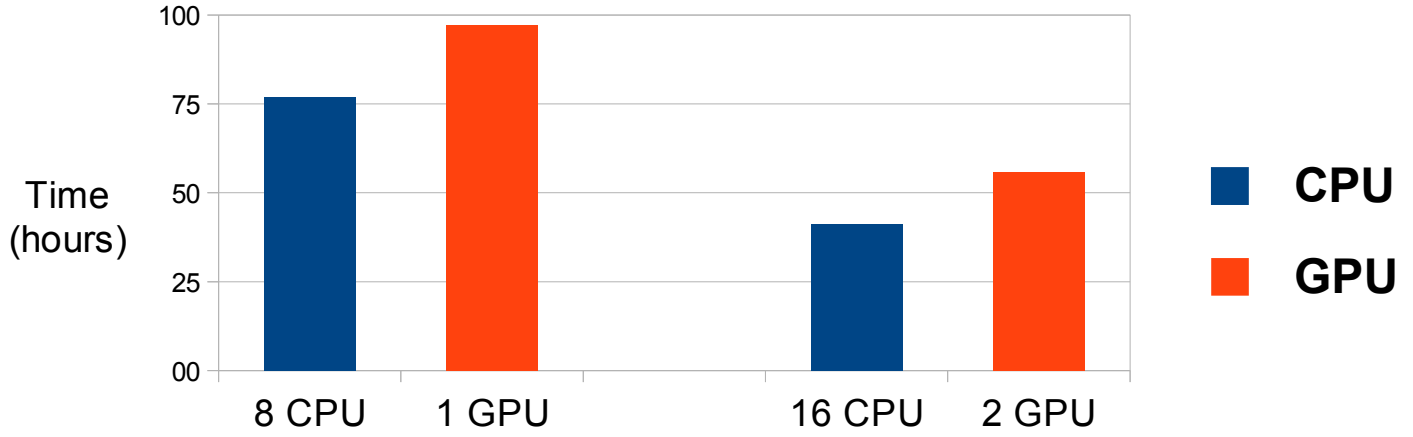


- Namd2 is 5 time faster with GPU's
- BigDFT is 8 time faster with GPU's

- Efficiency (Namd2)
  - Average GPU usage is less than 40%



  - 2 Quad Core are 1.3 time faster than 1 GPU



- Pros
  - GPU are very powerful
  - Reduce execution time
- Cons
  - GPU need highly parallel code to deliver speed
  - Program need to be ported or rewritten
  - User must learn a new language
  - Only suitable for massively parallel application

- Questions??